

# ACCELERATION OF PRIMAL-DUAL METHODS BY PRECONDITIONING AND SIMPLE SUBPROBLEM PROCEDURES\*

YANLI LIU<sup>†</sup>, YUNBEI XU<sup>‡</sup>, AND WOTAO YIN<sup>†</sup>

**Abstract.** Primal-Dual Hybrid Gradient (PDHG) and Alternating Direction Method of Multipliers (ADMM) are two widely-used first-order optimization methods. They reduce a difficult problem to simple subproblems, so they are easy to implement and have many applications. As first-order methods, however, they are sensitive to problem conditions and can struggle to reach the desired accuracy. To improve their performance, researchers have proposed techniques such as diagonal preconditioning and inexact subproblems. This paper realizes additional speedup about one order of magnitude.

Specifically, we choose non-diagonal preconditioners that are much more effective than diagonal ones. Because of this, we lose closed-form solutions to some subproblems, but we found simple procedures to replace them such as a few proximal-gradient iterations or a few epochs of proximal block-coordinate descent, which are in closed forms. We show global convergence while fixing the number of those steps in every outer iteration. Therefore, our method is reliable and straightforward.

Our method opens the choices of preconditioners and maintains both low per-iteration cost and global convergence. Consequently, on several typical applications of primal-dual first-order methods, we obtain 4–95× speedup over the existing state-of-the-art.

**Key words.** Primal-Dual Hybrid Gradient, Alternating Direction Method of Multipliers, preconditioning, fixed number of inner iterations, structured subproblem

**AMS subject classifications.** 49M29, 65K10, 65Y20, 90C25

**1. Introduction.** In this paper, we consider the following optimization problem:

$$(1.1) \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + g(Ax),$$

together with its dual problem:

$$(1.2) \quad \underset{z \in \mathbb{R}^m}{\text{minimize}} \quad f^*(-A^T z) + g^*(z),$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  and  $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  are closed proper convex, and  $A \in \mathbb{R}^{m \times n}$  is a matrix,  $f^*$  and  $g^*$  are the convex conjugates of  $f$  and  $g$ , respectively.

Formulations (1.1) or (1.2) are abstractions of many application problems, which include image restoration [43], magnetic resonance imaging [39], network optimization [16], computer vision [33], and earth mover’s distance [25].

To solve (1.1), one can apply primal-Dual algorithms such as Primal-Dual Hybrid Gradient (PDHG) and Alternating Direction Method of Multipliers (ADMM). However, as first-order algorithms, PDHG and ADMM suffer from slow (tail) convergence especially on poorly conditioned problems, when they may take thousands of iterations and still struggle reaching just four digits of accuracy. While they have many other advantages such as being easy to implement and friendly to parallelization, having their performance very sensitive to problem conditions is their main disadvantage. To improve the performance of PDHG and ADMM, researchers have tried using preconditioners, but for reasons we discuss below, only diagonal preconditioners so

---

\*Submitted to the editors XXX 2018.

**Funding:** The work of Y. Liu and W. Yin is supported in part by NSF DMS-1720237 and ONR N000141712162.

<sup>†</sup>Department of Mathematics, University of California, Los Angeles, CA ([yanli@math.ucla.edu](mailto:yanli@math.ucla.edu), [wotaoyin@math.ucla.edu](mailto:wotaoyin@math.ucla.edu)).

<sup>‡</sup>Graduate School of Business, Columbia University, New York, NY ([yunbei.xu@gsb.columbia.edu](mailto:yunbei.xu@gsb.columbia.edu)).

far. Depending on the application and how one applies splitting, PDHG and ADMM may or may not have subproblems with closed-form solutions. When they do not, researchers have studied approximate subproblem solutions to reduce the total running time. In the next subsection, we review the relevant works of preconditioning and inexact subproblems.

**1.1. Background.** Many problems to which we apply PDHG have separable functions  $f$  or  $g$ , or both, so the resulting PDHG subproblems often (though not always) have closed-form solutions. When subproblems are simple, we care mainly about the convergence rate of PDHG, which depends on the problem conditioning. To accelerate PDHG, diagonal preconditioning [32] was proposed since its diagonal structure maintains closed-form solutions for the subproblems and, therefore, reduces iteration complexity without making each iteration more difficult. In comparison, non-diagonal preconditioners are much more effective at reducing iteration complexity, but their off-diagonal entries couple different components in the subproblems, causing the loss of closed-form solutions of subproblems. So, it may appear we cannot have both fewer iterations and simple subproblems at the same time.

When a PDHG subproblem has no closed-form solution, one often uses an iterative algorithm to approximately solve it. We call it Inexact PDHG. Under certain conditions, Inexact PDHG still converges to the exact solution. Specifically, [34] uses three different types of conditions to skillfully control the errors of the subproblems; all those errors need to be summable over all the iterations and thereby requiring the error to diminish asymptotically. In an interesting method from [5, 6], one subproblem computes a proximal operator of a convex quadratic function, which can include a preconditioner and still has a closed-form solution involving matrix inversion. This proximal operator is successively applied  $n$  times in each iteration, for  $n \geq 1$ .

ADMM has different subproblems. One of its subproblems minimizes the sum of  $f(x)$  and a squared term involving  $Ax$ . Only when  $A$  has special structures does the subproblem have closed-form solutions. Inexact ADMM refers to the ADMM with at least one of its subproblems inexactly solved. An *absolute error criterion* was introduced in [12], where the subproblem errors are controlled by a summable (thus diminishing) sequence of error tolerances. To simplify the choice of the sequences, a *relative error criterion* was adopted in several later works, where the subproblem errors are controlled by a single parameter multiplying certain quantities that one can compute during the iterations. In [29], the parameters need to be square summable. In [24], the parameters are constants when both objective functions are Lipschitz differentiable. In [14, 13], two possible outcomes of the algorithm are described: (i) infinite outer loops and finite inner loops, and (ii) finite outer loops and the last inner loop is infinite, both guaranteeing convergence to a solution. On the other hand, it is unclear how to compare them. Since there is no bound on the number of inner loops in case (i), one may recognize it as case (ii) and stop the algorithm before it converges.

There are works that apply certain kinds of preconditioning to accelerate ADMM. Paper [18] uses diagonal preconditioning and observes improved performance. After that, non-diagonal preconditioning is analyzed [5, 6], which presents effective preconditioners for specific applications. One of their preconditioners needs to be inverted (though not needed in our method). Recently, preconditioning for strongly convex problems has also been studied [19] and promising numerical performances.

**1.2. Contributions.** Simply speaking, we find a way to have both non-diagonal preconditioners (thus much fewer iterations) and very simple subproblem procedures.

Our exposition takes a few steps. First, we present Preconditioned PDHG

(PrePDHG) and discuss how to choose preconditioners by minimizing an upper bound in PrePDHG’s ergodic convergence analysis. We can observe ADMM as a special case of PrePDHG where one of the preconditioners is identity (no preconditioning) and the other is the optimal choice, which minimizes the bound, and, thereby, explaining why ADMM often takes fewer iterations than PDHG.

Then, we show that PrePDHG still converges when one of its subproblems is solved inexactly to a specified condition. Remarkably, we do not need to verify this condition to stop a procedure since it is automatically satisfied as long as one applies a common iterative method for a *fixed number* of iterations. Common choices of subproblem procedures include proximal gradient descent, FISTA with restart, proximal block coordinate descent, and accelerated block-coordinate-gradient-descent (BCGD) methods (e.g., [27, 1, 21]). We call this method iPrePDHG (i for “inexact”).

We leave the other subproblem exactly solved in iPrePDHG since we have not encountered interesting applications that require non-diagonal preconditioners for both subproblems yet. If one is encountered, we can always split it in a way such that all ill-conditioned terms are collected in one subproblem.

Next, we apply iPrePDHG and develop effective preconditioners for a set of classic and representative applications of primal-dual splitting methods: image denoising, graph cut, optimal transport, and CT reconstruction. The CT reconstruction application uses a diagonal preconditioner in one subproblem, which has a closed-form solution, and a non-diagonal preconditioner in the other, which has no closed-form solution. In each of the other applications, one subproblem uses no preconditioner, and the other uses a non-diagonal preconditioner.

Finally, we numerically evaluated the performance of iPrePDHG using our recommended preconditioners. We obtained speedups of 4–95 times over the existing state-of-the-art. We believe it is a sufficient demonstration on how to apply preconditioners effectively and efficiently in PDHG.

Since we show ADMM is a special PrePDHG, our method also applies to ADMM. In fact, the iPrePDHG algorithms for three of the four applications are also Inexact Preconditioned ADMM under simple transformations.

**1.3. Organization.** The rest of this paper is organized as follows: Section 2 establishes notation and reviews basics. In the first part of Section 3, we provide a criterion for choosing preconditioners. In its second part, we introduce the condition for inexact subproblems, which can be automatically satisfied by iterating a fixed number of certain inner loops. This method is iPrePDHG. In the last part of Section 3, we establish the convergence of iPrePDHG. Section 4 describes specific preconditioners and reports numerical results. Finally, Section 5 concludes the paper.

**2. Preliminaries.** In this section, we introduce our notation and state the basic results that we need later.

We use  $\|\cdot\|$  for  $\ell_2$ -norm and  $\langle \cdot, \cdot \rangle$  for dot product.  $M \succ 0$  means  $M$  is a symmetric, positive definite matrix, and  $M \succeq 0$  means  $M$  is a symmetric, positive semidefinite matrix.

We write  $\lambda_{\min}(M)$  and  $\lambda_{\max}(M)$  as the smallest and the largest eigenvalues of  $M$ , respectively, and  $\kappa(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}$  as the condition number of  $M$ . For  $M \succeq 0$ , let  $\|\cdot\|_M$  and  $\langle \cdot, \cdot \rangle_M$  denote the semi-norm and inner product induced by  $M$ , respectively. If  $M \succ 0$ ,  $\|\cdot\|_M$  is a norm.

For a proper closed convex function  $\phi : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ , its subdifferential at

$x \in \mathbf{dom}f$  is written as

$$\partial\phi(x) = \{v \in \mathbb{R}^n \mid \phi(z) \geq \phi(x) + \langle v, z - x \rangle \ \forall z \in \mathbb{R}^n\},$$

and its convex conjugate as

$$\phi^*(y) = \sup_{x \in \mathbb{R}^n} \{\langle y, x \rangle - \phi(x)\}.$$

We have  $y \in \partial\phi(x)$  if and only if  $x \in \partial\phi^*(y)$ .

For any  $M \succ 0$ , we define the extended proximal operator of  $\phi$  as

$$(2.1) \quad \text{Prox}_\phi^M(x) := \arg \min_{y \in \mathbb{R}^n} \{\phi(y) + \frac{1}{2} \|y - x\|_M^2\}.$$

If  $M = \gamma^{-1}I$  for  $\gamma > 0$ , it reduces to a classic proximal operator.

We also have the following generalization of Moreau's Identity:

LEMMA 2.1 ([10], Theorem 3.1(ii)). *For any proper closed convex function  $\phi$  and  $M \succ 0$ , we have*

$$(2.2) \quad x = \text{Prox}_\phi^M(x) + M^{-1} \text{Prox}_{\phi^*}^{M^{-1}}(Mx).$$

We say a proper closed function is a Kurdyka-Łojasiewicz (KL) function if, for each  $x_0 \in \mathbf{dom}f$ , there exist  $\eta \in (0, \infty]$ , a neighborhood  $U$  of  $x_0$ , and a continuous concave function  $\varphi : [0, \eta) \rightarrow \mathbb{R}_+$  such that:

1.  $\varphi(0) = 0$ ,
2.  $\varphi$  is  $C^1$  on  $(0, \eta)$ ,
3. for all  $s \in (0, \eta)$ ,  $\varphi'(s) > 0$ ,
4. for all  $x \in U \cap \{x \mid f(x_0) < f(x) < f(x_0) + \eta\}$ , the KL inequality holds:

$$\varphi'(f(x) - f(x_0)) \text{dist}(0, \partial f(x)) \geq 1.$$

**3. Main results.** This section presents our results on preconditioner selection, the proposed method iPrePDHG, and its convergence.

Throughout this section, we assume the following regularity assumptions:

ASSUMPTION 1.

1.  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ ,  $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  are proper closed convex.
2. A primal-dual solution pair  $(x^*, z^*)$  of (1.1) and (1.2) exists, i.e.,

$$\mathbf{0} \in \partial f(x^*) + A^T z^*, \quad \mathbf{0} \in \partial g(Ax^*) - z^*.$$

The problem (1.1) also has the following convex-concave saddle-point formulation:

$$(3.1) \quad \min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}^m} \varphi(x, z) := f(x) + \langle Ax, z \rangle - g^*(z).$$

A primal-dual solution pair  $(x^*, z^*)$  is a solution of (3.1).

**3.1. Preconditioned PDHG.** The method of Primal-Dual Hybrid Gradient or PDHG [43, 7] for solving (1.1) refers to the iteration

$$(3.2) \quad \begin{aligned} x^{k+1} &= \text{Prox}_{\tau f}(x^k - \tau A^T z^k), \\ z^{k+1} &= \text{Prox}_{\sigma g^*}(z^k + \sigma A(2x^{k+1} - x^k)). \end{aligned}$$

When  $\frac{1}{\tau\sigma} \geq \|A\|^2$ , the iterates of (3.2) converge [7] to a primal-dual solution pair of (1.1). We can generalize (3.2) by applying preconditioners  $M_1, M_2 \succ 0$  (their choices are discussed below) to obtain Preconditioned PDHG or PrePDHG:

$$(3.3) \quad \begin{aligned} x^{k+1} &= \text{Prox}_f^{M_1}(x^k - M_1^{-1}A^T z^k), \\ z^{k+1} &= \text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}A(2x^{k+1} - x^k)), \end{aligned}$$

where the extended proximal operators  $\text{Prox}_f^{M_1}$  and  $\text{Prox}_{g^*}^{M_2}$  are defined in (2.1). We can obtain the convergence of PrePDHG using the analysis in [8].

There is no need to compute  $M_1^{-1}$  and  $M_2^{-1}$  since (3.3) is equivalent to

$$(3.4) \quad \begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \{f(x) + \langle x - x^k, A^T z^k \rangle + \frac{1}{2} \|x - x^k\|_{M_1}^2\}, \\ z^{k+1} &= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) - \langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{1}{2} \|z - z^k\|_{M_2}^2\}. \end{aligned}$$

**3.2. Choice of preconditioners.** In this section, we discuss how to select appropriate preconditioners  $M_1$  and  $M_2$ . As a by-product, we show that ADMM corresponds to choosing  $M_1 = \frac{1}{\tau} I_{n \times n}$  and optimally choosing  $M_2 = \tau A A^T$ , thereby, explaining why ADMM appears to be faster than PDHG.

Let us start with the following lemma, which characterizes primal-dual solution pairs of (1.1) and (1.2).

LEMMA 3.1. *Under Assumption 1,  $(X, Z)$  is a primal-dual solution pair of (1.1) if and only if  $\varphi(X, z) - \varphi(x, Z) \leq 0$  for any  $(x, z) \in \mathbb{R}^{n+m}$ , where  $\varphi$  is given in the saddle-point formulation (3.1).*

*Proof.* If  $(X, Z)$  is a primal-dual solution pair of (1.1), then

$$-A^T Z \in \partial f(X), \quad AX \in \partial g^*(Z).$$

Hence, for any  $(x, z) \in \mathbb{R}^{n+m}$  we have

$$f(x) \geq f(X) + \langle -A^T Z, x - X \rangle, \quad g^*(z) \geq g^*(Z) + \langle AX, z - Z \rangle.$$

Adding them together yields  $\varphi(X, z) - \varphi(x, Z) \leq 0$ .

On the other hand, if  $\varphi(X, z) - \varphi(x, Z) \leq 0$  for any  $(x, z) \in \mathbb{R}^{n+m}$ , then

$$\langle AX, z \rangle + f(X) - g^*(z) - \langle Ax, Z \rangle - f(x) + g^*(Z) \leq 0 \quad \text{for any } (x, z) \in \mathbb{R}^{n+m}.$$

Taking  $x = X$  yields  $\langle AX, z - Z \rangle - g^*(z) + g^*(Z) \leq 0$ , so  $AX \in \partial g^*(Z)$ ; Similarly, taking  $z = Z$  gives  $\langle AX - Ax, Z \rangle + f(X) - f(x) \leq 0$ , so  $-A^T Z \in \partial f(X)$ . As a result,  $(X, Z)$  is a primal-dual solution pair of (1.1).  $\square$

We present the following convergence result, adapted from Theorem 1 of [8].

THEOREM 3.2. *Let  $(x^k, z^k), k = 0, 1, \dots, N$  be a sequence generated by PrePDHG (3.3). Under Assumption 1, if in addition*

$$(3.5) \quad \tilde{M} := \begin{pmatrix} M_1 & -A^T \\ -A & M_2 \end{pmatrix} \succeq 0,$$

then, for any  $x \in \mathbb{R}^n$  and  $z \in \mathbb{R}^m$ , it holds that

$$(3.6) \quad \varphi(X^N, z) - \varphi(x, Z^N) \leq \frac{1}{2N} (x - x^0, z - z^0) \begin{pmatrix} M_1 & -A^T \\ -A & M_2 \end{pmatrix} \begin{pmatrix} x - x^0 \\ z - z^0 \end{pmatrix},$$

where  $X^N = \frac{1}{N} \sum_{i=1}^N x^i$  and  $Z^N = \frac{1}{N} \sum_{i=1}^N z^i$ .

*Proof.* This follows from Theorem 1 of [8] by setting  $L_f = 0$ ,  $\frac{1}{\tau}D_x(x, x_0) = \frac{1}{2}\|x - x^0\|_{M_1}^2$ ,  $\frac{1}{\sigma}D_z(z, z_0) = \frac{1}{2}\|z - z^0\|_{M_2}^2$ , and  $K = A$ . Note that in Remark 1 of [8],  $D_x$  and  $D_z$  need to be 1-strongly convex to ensure that inequality (13) holds, which is exactly our (3.5). Therefore, we don't need  $D_x$  and  $D_z$  to be strongly convex.  $\square$

Based on the above results, one approach to accelerate convergence is to choose preconditioners  $M_1$  and  $M_2$  to obey (3.5) and minimize the right-hand side of (3.6). When a pair of preconditioner matrices attains this minimum, we say they are optimal. When one of them is fixed, the other that attains the minimum is also called optimal.

By Schur complement, the condition (3.5) is equivalent to  $M_2 \succeq AM_1^{-1}A^T$ . Hence, for any given  $M_1 \succ 0$ , the optimal  $M_2$  is  $AM_1^{-1}A^T$ .

Original PDHG (3.2) corresponds to  $M_1 = \frac{1}{\tau}I_{n \times n}$ ,  $M_2 = \frac{1}{\sigma}I_{m \times m}$  with  $\tau$  and  $\sigma$  obeying  $\frac{1}{\tau\sigma} \geq \|A\|^2$  for convergence. In Appendix A, we show that ADMM for problem (1.1) corresponds to setting  $M_1 = \frac{1}{\tau}I_{n \times n}$ ,  $M_2 = \tau AA^T$ ,  $M_2$  is optimal since  $AM_1^{-1}A^T = \tau AA^T = M_2$ . (This is related to, but different from, the result in [7, Sec. 4.3] stating that PDHG is equivalent to a preconditioned ADMM.)

By using more general pairs of  $M_1, M_2$ , we can potentially have even fewer iterations of PrePDHG than ADMM.

**3.3. PrePDHG with fixed inner iterations.** It wastes total time to solve the subproblems in (3.4) very accurately. It is more efficient to develop a proper condition and stop the subproblem procedure, which we call *inner iterations*, once the condition is satisfied. It is even better if we can simply fix the number of inner iterations and still guarantee global convergence.

In this subsection, we describe the ‘‘bounded relative error’’ of the  $z$ -subproblem in (3.3) and then show that this can be satisfied by running a fixed number of inner iterations, uniformly for every outer loop.

**DEFINITION 3.3** (Bounded relative error condition). *Given  $x^k, x^{k+1}$  and  $z^k$ , we say that the  $z$ -subproblem in PrePDHG (3.3) is solved to a bounded relative error if there is a constant  $c > 0$  such that*

$$(3.7) \quad \mathbf{0} \in \partial g^*(z^{k+1}) + M_2(z^{k+1} - z^k - M_2^{-1}A(2x^{k+1} - x^k)) + \varepsilon^{k+1},$$

$$(3.8) \quad \|\varepsilon^{k+1}\| \leq c\|z^{k+1} - z^k\|.$$

Remarkably, this condition does not need to be checked at run time. For a fixed  $c > 0$ , the condition can be satisfied by a fixed number of inner iterations using, for example, proximal gradient iteration (Theorem 3.4). One can also use faster solvers, e.g., FISTA with restart [30], and solvers that suit the subproblem structure, e.g., cyclic proximal BCD (Theorem 3.5). Although the error in solving  $z$ -subproblems appears to be neither summable nor square summable, convergence can still be established. But first, we summarize this method in Algorithm 3.1.

**THEOREM 3.4.** *Take Assumption 1. Suppose in iPrePDHG, or Algorithm 3.1, we choose  $S$  as the proximal-gradient step with stepsize  $\gamma \in (0, \frac{2\lambda_{\min}(M_2)}{\lambda_{\max}^2(M_2)})$  and repeat it  $p$  times, where  $p \geq 1$ . Then,  $z^{k+1} = z_p^{k+1}$  is an approximate solution to the  $z$ -subproblem up to a bounded relative error in (3.8) for*

$$(3.9) \quad c = c(p) = \frac{\frac{1}{\gamma} + \lambda_{\max}(M_2)}{1 - \tau^p} (\tau^p + \tau^{p-1}),$$

where  $\tau = \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))} < 1$ .

**Algorithm 3.1** Inexact Preconditioned PDHG or iPrePDHG

**Input:**  $f, g, A$  in (1.1), preconditioners  $M_1$  and  $M_2$ , initial  $(x_0, z_0)$ ,  $z$ -subproblem iterator  $S$ , inner iteration number  $p$ , max outer iteration number  $K$ .

**Output:**  $(x^K, z^K)$

---

```

1: for  $k \leftarrow 0, 1, \dots, K - 1$  do
2:    $x^{k+1} = \text{Prox}_f^{M_1}(x^k - M_1^{-1}A^T z^k)$ ;
3:    $z_0^{k+1} = z^k$ ;
4:   for  $i \leftarrow 0, 1, \dots, p - 1$  do
5:      $z_{i+1}^{k+1} = S(z_i^{k+1}, x^{k+1}, x^k)$ ;
6:   end for
7:    $z^{k+1} = z_p^{k+1}$ ;            $\triangleright$  which approximates  $\text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}A(2x^{k+1} - x^k))$ 
8: end for

```

---

*Proof.* The  $z$ -subproblem in (3.4) is of the form

$$(3.10) \quad \underset{z \in \mathbb{R}^m}{\text{minimize}} \quad h_1(z) + h_2(z),$$

for  $h_1(z) = g^*(z)$  and  $h_2(z) = \frac{1}{2}\|z - z^k - M_2^{-1}A(2x^{k+1} - x^k)\|_{M_2}^2$ . With our choice of  $S$  as the proximal-gradient descent step, the inner iterations are

$$(3.11) \quad \begin{aligned} z_0^{k+1} &= z^k, \\ z_{i+1}^{k+1} &= \text{Prox}_{\gamma h_1}(z_i^{k+1} - \gamma \nabla h_2(z_i^{k+1})), \quad i = 0, 1, \dots, p-1, \end{aligned}$$

Concerning the last iterate  $z^{k+1} = z_p^{k+1}$ , we have from the definition of  $\text{Prox}_{\gamma h_1}$  that

$$\mathbf{0} \in \partial h_1(z_p^{k+1}) + \nabla h_2(z_p^{k+1}) + \frac{1}{\gamma}(z_p^{k+1} - z_{p-1}^{k+1}).$$

Compare this with (3.7) and use  $z^{k+1} = z_p^{k+1}$  to get

$$\varepsilon^{k+1} = \frac{1}{\gamma}(z_p^{k+1} - z_{p-1}^{k+1}) + \nabla h_2(z_p^{k+1}) - \nabla h_2(z_{p-1}^{k+1}).$$

It remains to show that  $\varepsilon^{k+1}$  satisfies (3.8).

Let  $z_\star^{k+1}$  be the solution of (3.10),  $\alpha = \lambda_{\min}(M_2)$ , and  $\beta = \lambda_{\max}(M_2)$ . Then  $h_1(z)$  is convex and  $h_2(z)$  is  $\alpha$ -strongly convex and  $\beta$ -Lipschitz differentiable. Consequently, [3, Prop. 26.16(ii)] gives

$$\|z_i^{k+1} - z_\star^{k+1}\| \leq \tau^i \|z_0^{k+1} - z_\star^{k+1}\|, \quad \forall i = 0, 1, \dots, p,$$

where  $\tau = \sqrt{1 - \gamma(2\alpha - \gamma\beta^2)}$ .

Let  $a_i = \|z_i^{k+1} - z_\star^{k+1}\|$ . Then,  $a_i \leq \tau^i a_0$ . We can derive

$$(3.12) \quad \|\varepsilon^{k+1}\| \leq \left(\frac{1}{\gamma} + \beta\right) \|z_p^{k+1} - z_{p-1}^{k+1}\| \leq \left(\frac{1}{\gamma} + \beta\right)(a_p + a_{p-1}) \leq \left(\frac{1}{\gamma} + \beta\right)(\tau^p + \tau^{p-1})a_0.$$

On the other hand, we have

$$(3.13) \quad \|z^{k+1} - z^k\| \geq a_0 - a_p \geq (1 - \tau^p)a_0.$$

Combining these two equations yields

$$\|\varepsilon^{k+1}\| \leq c\|z^{k+1} - z^k\|,$$

where  $c$  is given in (3.9).  $\square$

Theorem 3.4 uses the iterator  $S$  that is the proximal-gradient step. It is straightforward to extend its proof to  $S$  being the FISTA step. We omit the proof.

In our next theorem, we let  $S$  be the iterator of one epoch of the cyclic proximal BCD method. A BCD method updates one block of coordinates at a time while fixing the remaining blocks. In one epoch of cyclic BCD, all the blocks of coordinates are sequentially updated, and every block is updated once. In cyclic *proximal* BCD, each block of coordinates is updated by a proximal-gradient step, just like (3.11) except only the chosen block is updated each time. When  $h_1$  is block separable, each update costs only a fraction of updating all the blocks together. When different blocks are updated one after another, the Gauss-Seidel effect brings more progress. In addition, since the Lipschitz constant of each block gradient of  $h_2$  is typically less than that of  $\nabla h_2$ , one can use a larger stepsize  $\gamma$  and get potentially even faster progress. Therefore, the iterator of cyclic proximal BCD is a better choice for  $S$ .

**THEOREM 3.5.** *Let Assumption 1 hold and  $g$  be block separable, i.e.,  $z = (z_1, z_2, \dots, z_l)$  and  $g(z) = \sum_{i=1}^l g_i(z_i)$ . Suppose in iPrePDHG, or Algorithm 3.1, we choose  $S$  as the iterator of cyclic proximal BCD with stepsize  $\gamma$  satisfying*

$$0 < \gamma \leq \min \left\{ \frac{2\lambda_{\min}(M_2)}{\lambda_{\max}^2(M_2)}, \frac{1 - \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))}}{4\sqrt{2}\gamma\lambda_{\max}(M_2)}, \frac{1}{4l\lambda_{\max}(M_2)}, \frac{2l\lambda_{\max}(M_2)}{17l\lambda_{\max}(M_2) + 2\left(\frac{1 - \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))}}{\gamma}\right)^2} \right\},$$

and we set  $p \geq 1$ . Then,  $z^{k+1} = z_p^{k+1}$  is an approximate solution to the  $z$ -subproblem up to a bounded relative error (3.8) for

$$(3.14) \quad c = c(p) = \frac{(l\lambda_{\max}(M_2) + \frac{1}{\gamma})(\rho^p + \rho^{p-1})}{1 - \rho^p},$$

where  $\rho = 1 - \frac{(1 - \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))})^2}{2\gamma} < 1$ .

*Proof.* See Appendix B.  $\square$

**3.4. Global convergence of iPrePDHG.** In this subsection, we proceed to establishing the convergence of Algorithm 3.1. Our approach first transforms Algorithm 3.1 into an equivalent algorithm in Proposition 3.6 below and then proves its convergence in Theorems 3.9 and 3.10 below.

First, let us show that PrePDHG (3.3) is equivalent to an algorithm applied on the dual problem (1.2). This equivalence is analogous to the equivalence between PDHG (3.2) and Linearized ADMM applied to the dual problem (1.2), shown in [15]. Specifically, PrePDHG is equivalent to

$$(3.15) \quad \begin{aligned} z^{k+1} &= \text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}AM_1^{-1}(-A^T z^k - y^k + u^k)), \\ y^{k+1} &= \text{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^{k+1}), \\ u^{k+1} &= u^k - A^T z^{k+1} - y^{k+1}. \end{aligned}$$



When  $M_1 = \frac{1}{\tau}I, M_2 = \lambda I$ , (3.15) reduces to Linearized ADMM, also known as Split Inexact Uzawa [42].

Furthermore, iPrePDHG in Algorithm 3.1 is equivalent to (3.15) with inexact subproblems, which we present in Algorithm 3.2.

---

**Algorithm 3.2** Inexact Preconditioned ADMM
 

---

**Input:**  $f^* : \mathbb{R}^n \rightarrow \mathbb{R}, g^* : \mathbb{R}^m \rightarrow \mathbb{R}, A \in \mathbb{R}^{m \times n}$ , preconditioners  $M_1$  and  $M_2$ , initial vector  $(z_0, y_0, u_0)$ , subproblem solver  $S$  for the  $z$ -subproblem in (3.15), number of inner loops  $p$ , number of outer iterations  $K$ .

**Output:**  $(z^K, y^K, u^K)$

```

1: for  $k \leftarrow 0, 1, \dots, K - 1$  do
2:    $z_0^{k+1} = z^k$ ;
3:   for  $i \leftarrow 0, 1, \dots, p - 1$  do
4:      $z_{i+1}^{k+1} = S(z_i^{k+1}, y^k, u^k)$ ;
5:   end for
6:    $z^{k+1} = z_p^{k+1}$ ;  $\triangleright$  approximate  $\text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}AM_1^{-1}(-A^T z^k - y^k + u^k))$ .
7:    $y^{k+1} = \text{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^{k+1})$ ;
8:    $u^{k+1} = u^k - A^T z^{k+1} - y^{k+1}$ ;
9: end for

```

---

Let us define the following generalized augmented Lagrangian for (3.15):

$$(3.16) \quad L(z, y, u) = g^*(z) + f^*(y) + \langle -A^T z - y, M_1^{-1}u \rangle + \frac{1}{2} \|A^T z + y\|_{M_1^{-1}}^2.$$

Inspired by [40], we use (3.16) as the Lyapunov function to establish convergence of Algorithm 3.2 and, equivalently, the convergence of Algorithm 3.1.

**PROPOSITION 3.6.** *Under Assumption 1 and the transforms  $u^k = M_1 x^k, y^{k+1} = u^k - A^T z^k - u^{k+1}$ , PrePDHG (3.3) is equivalent to (3.15), and iPrePDHG in Algorithm 3.1 is equivalent to Algorithm 3.2.*

*Proof.* Set  $u^k = M_1 x^k, y^{k+1} = u^k - A^T z^k - u^{k+1}$ . Then (2.2) and (3.3) yield

$$y^{k+1} = M_1 x^k - A^T z^k - M_1 x^{k+1} = \text{Prox}_{f^*}^{M_1^{-1}}(u^k - A^T z^k),$$

and

$$\begin{aligned} u^{k+1} &= u^k - A^T z^k - y^{k+1}, \\ z^{k+1} &= \text{Prox}_{g^*}^{M_2}(z^k + M_2^{-1}AM_1^{-1}(-A^T z^k - y^{k+1} + u^{k+1})). \end{aligned}$$

If the  $z$ -update is performed first, then we arrive at (3.15).

In iPrePDHG or Algorithm 3.1, we are solving the  $z$ -subproblem of PrePDHG (3.3) approximately to the bounded relative error in Definition 3.3. This is equivalent to doing the same to the  $z$ -subproblem of (3.15), which yields Algorithm 3.2.  $\square$

We establish convergence under the following additional assumptions.

**ASSUMPTION 2.**

1.  $f(x)$  is  $\mu_f$ -strongly convex.
2.  $g^*(z) + f^*(-A^T z)$  is coercive, i.e.,  $\lim_{\|z\| \rightarrow \infty} g^*(z) + f^*(-A^T z) = \infty$ .
3.  $g^*(z)$  is a KE function.

**THEOREM 3.7.** *Take Assumptions 1 and 2. Choose any preconditioners  $M_1, M_2$  and inner iteration number  $p$  such that*

$$(3.17) \quad C_1 = \frac{1}{2}M_1^{-1} - \frac{\|M_1\|}{\mu_f^2}I_{n \times n} \succ 0,$$

$$(3.18) \quad C_2 = M_2 - \frac{1}{2}AM_1^{-1}A^T - c(p)I_{m \times m} \succ 0,$$

where  $c(p)$  depends on the  $z$ -subproblem iterator  $S$  and  $M_2$  (e.g., (3.9) and (3.14)). Define  $L^k := L(z^k, y^k, u^k)$ . Then, Algorithm 3.2 satisfies the following sufficient descent and lower boundedness properties, respectively:

$$(3.19) \quad L^k - L^{k+1} \geq \|y^k - y^{k+1}\|_{C_1}^2 + \|z^k - z^{k+1}\|_{C_2}^2,$$

$$(3.20) \quad L^k \geq g^*(z^*) + f^*(-A^T z^*) > -\infty.$$

*Proof.* Since the  $z$ -subproblem of Algorithm 3.2 is solved to the bounded relative error in Def. 3.3, we have

$$(3.21) \quad \mathbf{0} \in \partial g^*(z^{k+1}) + M_2(z^{k+1} - z^k - M_2^{-1}AM_1^{-1}(-A^T z^k - y^k + u^k)) + \varepsilon^{k+1},$$

where  $\varepsilon^{k+1}$  satisfies (3.8):

$$(3.22) \quad \|\varepsilon^{k+1}\| \leq c(p)\|z^{k+1} - z^k\|.$$

The  $y$  and  $u$  updates produce

$$(3.23) \quad \mathbf{0} = \nabla f^*(y^{k+1}) + M_1^{-1}(y^{k+1} - u^k + A^T z^{k+1}) = \nabla f^*(y^{k+1}) - M_1^{-1}u^{k+1},$$

$$(3.24) \quad u^{k+1} = u^k - A^T z^{k+1} - y^{k+1}.$$

In order to show (3.19), let us write

$$\begin{aligned} g^*(z^k) &\geq g^*(z^{k+1}) \\ &\quad + \langle M_2(z^k - z^{k+1}) + AM_1^{-1}(-A^T z^k - y^k + u^k) - \varepsilon^{k+1}, z^k - z^{k+1} \rangle, \\ f^*(y^k) &\geq f^*(y^{k+1}) + \langle M_1^{-1}u^{k+1}, y^k - y^{k+1} \rangle, \end{aligned}$$

Assembling these inequalities with (3.22) gives us

$$\begin{aligned} L^k - L^{k+1} &\geq \|z^k - z^{k+1}\|_{M_2 - c(p)I_{m \times m}}^2 \\ &\quad + \langle AM_1^{-1}(-A^T z^k - y^k + u^k), z^k - z^{k+1} \rangle + \langle M_1^{-1}u^{k+1}, y^k - y^{k+1} \rangle \\ &\quad + \langle -A^T z^k - y^k, M_1^{-1}u^k \rangle - \langle A^T z^{k+1} - y^{k+1}, M_1^{-1}(u^k - A^T z^{k+1} - y^{k+1}) \rangle \\ &\quad + \frac{1}{2}\|A^T z^k + y^k\|_{M_1^{-1}}^2 - \frac{1}{2}\|A^T z^{k+1} + y^{k+1}\|_{M_1^{-1}}^2 \\ &= \|z^k - z^{k+1}\|_{M_2 - c(p)I_{m \times m}}^2 \\ (A) \quad &\quad + \langle AM_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \rangle + \langle M_1^{-1}u^{k+1}, y^k - y^{k+1} \rangle \\ (B) \quad &\quad + \langle -y^k, M_1^{-1}u^k \rangle - \langle -y^{k+1}, M_1^{-1}u^k \rangle \\ &\quad + \frac{1}{2}\|A^T z^k + y^k\|_{M_1^{-1}}^2 - \frac{3}{2}\|A^T z^{k+1} + y^{k+1}\|_{M_1^{-1}}^2, \end{aligned}$$

where the terms in (A) and (B) simplify to

$$(3.25) \quad \langle AM_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \rangle + \langle M_1^{-1}(-A^T z^{k+1} - y^{k+1}), y^k - y^{k+1} \rangle.$$

Apply the following cosine rule on the two inner products above:

$$\langle a - b, a - c \rangle_{M_1^{-1}} = \frac{1}{2} \|a - b\|_{M_1^{-1}}^2 + \frac{1}{2} \|a - c\|_{M_1^{-1}}^2 - \frac{1}{2} \|b - c\|_{M_1^{-1}}^2.$$

Set  $a = A^T z^k$ ,  $c = A^T z^{k+1}$ , and  $b = -y^k$  to obtain

$$\begin{aligned} \langle AM_1^{-1}(-A^T z^k - y^k), z^k - z^{k+1} \rangle &= -\frac{1}{2} \|A^T z^k + y^k\|_{M_1^{-1}}^2 - \frac{1}{2} \|A^T z^k - A^T z^{k+1}\|_{M_1^{-1}}^2 \\ (3.26) \quad &+ \frac{1}{2} \|y^k + A^T z^{k+1}\|_{M_1^{-1}}^2. \end{aligned}$$

Set  $a = y^{k+1}$ ,  $c = y^k$ , and  $b = -A^T z^{k+1}$  to obtain

$$\begin{aligned} \langle M_1^{-1}(-A^T z^{k+1} - y^{k+1}), y^k - y^{k+1} \rangle &= \frac{1}{2} \|A^T z^{k+1} + y^{k+1}\|_{M_1^{-1}}^2 + \frac{1}{2} \|y^k - y^{k+1}\|_{M_1^{-1}}^2 \\ (3.27) \quad &- \frac{1}{2} \|A^T z^{k+1} + y^k\|_{M_1^{-1}}^2. \end{aligned}$$

Combining (3.25), (3.26), and (3.27) yields

$$\begin{aligned} L^k - L^{k+1} &\geq \|z^k - z^{k+1}\|_{M_2 - \frac{1}{2} AM_1^{-1} A^T - c(p) I_{m \times m}}^2 + \|y^k - y^{k+1}\|_{\frac{1}{2} M_1^{-1}}^2 \\ (3.28) \quad &- \|A^T z^{k+1} + y^{k+1}\|_{M_1^{-1}}^2. \end{aligned}$$

Since  $f$  is  $\mu_f$ -strongly convex, we know that  $\nabla f^*$  is  $\frac{1}{\mu_f}$ -Lipschitz continuous. Consequently,

$$\begin{aligned} \|A^T z^{k+1} + y^{k+1}\|_{M_1^{-1}}^2 &= \|u^k - u^{k+1}\|_{M_1^{-1}}^2 \leq \frac{1}{\lambda_{\min}(M_1^{-1})} \|M_1^{-1}(u^k - u^{k+1})\|^2 \\ (3.29) \quad &\stackrel{(3.23)}{\leq} \frac{\|M_1\|}{\mu_f^2} \|y^k - y^{k+1}\|^2. \end{aligned}$$

Combining (3.28) and (3.29) gives us (3.19).

Now, to show (3.20), we use (3.23) and smoothness of  $f^*$  to get

$$f^*(y^k) \geq f^*(-A^T z^k) + \langle M_1^{-1} u^k, y^k + A^T z^k \rangle - \frac{1}{2\mu_f} \|A^T z^k + y^k\|^2.$$

Hence, we arrive at

$$\begin{aligned} L^k &= g^*(z^k) + f^*(y^k) + \langle -A^T z^k - y^k, M_1^{-1} u^k \rangle + \frac{1}{2} \|A^T z^k + y^k\|_{M_1^{-1}}^2 \\ (3.30) \quad &\geq g^*(z^k) + f^*(-A^T z^k) + \frac{1}{2} \|A^T z^k + y^k\|_{M_1^{-1}}^2 - \frac{1}{2\mu_f} \|A^T z^k + y^k\|^2. \end{aligned}$$

Since  $C_1 > 0$  if and only if  $\mu_f > \sqrt{2}\|M_1\|$ , (3.20) follows.  $\square$

Next, we provide a simple choice of  $M_1, M_2$ , and  $p$  that ensures the positive definiteness of  $C_1$  and  $C_2$  in Theorem 3.7.

**LEMMA 3.8.** *In order to ensure (3.17) and (3.18), it suffices to set  $M_1 = \frac{1}{\tau} I_{n \times n}$  where  $\tau < \frac{1}{\sqrt{2}} \mu_f$ ,  $M_2 = \tau A A^T + \gamma I_{m \times m}$  with any  $\gamma > 0$ , and  $p$  is large enough such that  $c(p) < \gamma$ .*

*Proof.* Since  $M_1 = \frac{1}{\tau}I_{n \times n}$ , it is evident that  $C_1 \succ 0$  if and only if  $\tau < \frac{1}{\sqrt{2}}\mu_f$ . With  $M_1 = \frac{1}{\tau}I_{n \times n}$  and  $M_2 = \tau AA^T + I_{m \times m}$ , we have

$$C_2 = \frac{1}{2}\tau AA^T + (\gamma - c(p))I_{m \times m},$$

since  $c(p) \propto \alpha^p$  for some  $0 < \alpha < 1$ , we know that there exists  $p_0$  such that  $C_2 \succ 0$  for any  $p \geq p_0$ .  $\square$

We conclude this section by showing that  $(x^k, z^k)$  in Algorithm 3.1 converges subsequentially to a primal-dual solution pair of (1.1) and (1.2).

**THEOREM 3.9.** *Let the assumptions in Theorem 3.7 hold. Then,  $(x^k, z^k)$  in Algorithm 3.1 is bounded, and any cluster point of  $\{x^k, z^k\}$  is a primal-dual solution pair of (1.1) and (1.2).*

*Proof.* According to Theorem 3.6, it is sufficient to show that  $\{M_1^{-1}u^k, z^k\}$  is bounded, and its cluster points are primal-dual solution pairs of (1.1).

Since  $L^k$  is nonincreasing, (3.30) tells us that

$$g^*(z^k) + f^*(-A^T z^k) + \frac{1}{2}\|A^T z^k + y^k\|_{M_1^{-1}}^2 \leq L^0 < +\infty.$$

Since  $g^*(z) + f^*(-A^T z)$  is coercive,  $\{z^k\}$  is bounded, and, by the boundedness of  $\{A^T z^k + y^k\}$ ,  $\{y^k\}$  is also bounded. Furthermore, (3.23) gives us

$$\|M_1^{-1}(u^k - u^0)\| \leq \frac{1}{\mu_f}\|y^k - y^0\|.$$

Therefore,  $\{M_1^{-1}u^k\}$  is bounded, too.

Let  $(z^c, y^c, u^c)$  be a cluster point of  $\{z^k, y^k, u^k\}$ . We shall show  $(z^c, y^c, u^c)$  is a saddle point of  $L(z, y, u)$ , i.e.,

$$(3.31) \quad \mathbf{0} \in \partial L(z^c, y^c, u^c),$$

or equivalently,

$$\begin{aligned} \mathbf{0} &\in \partial g^*(z^c) - AM_1^{-1}u^c, \\ \mathbf{0} &= \nabla f^*(y^c) - M_1^{-1}u^c, \\ \mathbf{0} &= A^T z^c + y^c, \end{aligned}$$

which ensures  $(M_1^{-1}u^c, z^c)$  to be a primal-dual solution pair of (1.1).

In order to show (3.31), we first notice that (3.16) gives

$$\begin{aligned} \partial_x L(z^{k+1}, y^{k+1}, u^{k+1}) &= \partial g^*(z^{k+1}) - AM_1^{-1}u^{k+1} + AM_1^{-1}(A^T z^{k+1} + y^{k+1}), \\ \nabla_y L(z^{k+1}, y^{k+1}, u^{k+1}) &= \nabla f^*(y^{k+1}) - M_1^{-1}u^{k+1} + M_1^{-1}(A^T z^{k+1} + y^{k+1}), \\ \nabla_u L(z^{k+1}, y^{k+1}, u^{k+1}) &= M_1^{-1}(-A^T z^{k+1} - y^{k+1}). \end{aligned}$$

Comparing these with the optimality conditions (3.21), (3.23), and (3.24), we have

$$d^{k+1} = (d_z^{k+1}, d_y^{k+1}, d_u^{k+1}) \in \partial L(z^{k+1}, y^{k+1}, u^{k+1}),$$

where

$$\begin{aligned} d_z^{k+1} &= M_2(z^k - z^{k+1}) + 2AM_1^{-1}(u^k - u^{k+1}) - AM_1^{-1}(u^{k-1} - u^k) - \varepsilon^{k+1}, \\ d_y^{k+1} &= M_1^{-1}(u^k - u^{k+1}), \\ d_u^{k+1} &= M_1^{-1}(u^{k+1} - u^k). \end{aligned}$$

Since (3.19) and (3.20) imply  $z^k - z^{k+1}, y^k - y^{k+1} \rightarrow \mathbf{0}$ , (3.23) gives  $u^k - u^{k+1} \rightarrow \mathbf{0}$ . Combine these with (3.8), we have  $d^k \rightarrow \mathbf{0}$ .

Finally, let us take a subsequence  $\{z^{k_s}, y^{k_s}, u^{k_s}\} \rightarrow (z^c, y^c, u^c)$ . Since  $d^{k_s} \rightarrow \mathbf{0}$  as  $s \rightarrow +\infty$ , [36, Def. 8.3] and [36, Prop. 8.12] yield (3.31), which tells us that  $(M_1^{-1}u^c, z^c)$  is a primal-dual solution pair of (1.1).  $\square$

We can show that the whole sequence  $(x^k, z^k)$  in Algorithm 3.1 converges. Since the proof consists of a standard technique of using the KŁ property in Assumption 2, which is not relevant to the main idea of this subsection, we leave it to Appendix C.

**THEOREM 3.10.** *Let the assumptions in Theorem 3.7 hold. Then,  $\{x^k, z^k\}$  in Algorithm 3.1 converges to a primal-dual solution pair of (1.1).*

*Proof.* See Appendix C.  $\square$

**4. Numerical experiments.** In this section, we compare our iPrePDHG (Algorithm 3.1) with (original) PDHG (3.2) and diagonally-preconditioned PDHG (DP-PDHG) [32]. We consider four popular applications of PDHG: TV-L<sup>1</sup> denoising, graph cuts, estimation of earth mover's distance, and CT reconstruction.

For the preconditioners  $M_1$  and  $M_2$  in iPrePDHG, We will choose  $M_1 = \frac{1}{\tau}I_{n \times n}$  and  $M_2 = \tau AA^T$ , which corresponds to ADMM and  $M_2$  is optimal (see subsection 3.2). The number of inner loops  $p$  is taken from  $\{1, 2, 3\}$ . Although these choices may be more aggressive than what's presented in Lemma 3.8 and  $f$  may not be strongly convex in our experiments, we still observe significant speedup compared to other algorithms.

When we write these examples in the form of (1.1), the matrix  $A$  (or a part of  $A$ ) is one of the following operators:

**Case 1:** 2D discrete gradient operator  $D : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{2M \times N}$ :

For images of size  $M \times N$  and grid step size  $h$ , we have.

$$(Du)_{i,j} = \begin{pmatrix} (Du)_{i,j}^1 \\ (Du)_{i,j}^2 \end{pmatrix},$$

where

$$\begin{aligned} (Du)_{i,j}^1 &= \begin{cases} \frac{1}{h}(u_{i+1,j} - u_{i,j}) & \text{if } i < M, \\ 0 & \text{if } i = M, \end{cases} \\ (Du)_{i,j}^2 &= \begin{cases} \frac{1}{h}(u_{i,j+1} - u_{i,j}) & \text{if } j < N, \\ 0 & \text{if } j = N. \end{cases} \end{aligned}$$

**Case 2:** Weighted gradient operator  $D_w : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{2M \times N}$ :

$$D_w = \text{diag}(w)D,$$

where  $w \in (\mathbb{R}^+)^{2MN}$  is a weight vector.

**Case 3:** 2D discrete divergence operator:  $\text{div}: \mathbb{R}^{2M \times N} \rightarrow \mathbb{R}^{M \times N}$  given by

$$(4.1) \quad \text{div}(p)_{i,j} = h(p_{i,j}^1 - p_{i-1,j}^1 + p_{i,j}^2 - p_{i,j-1}^2),$$

where  $p = (p^1, p^2)^T \in \mathbb{R}^{2M \times N}$ ,  $p_{0,j}^1 = p_{M,j}^1 = 0$  and  $p_{i,0}^2 = p_{i,N}^2 = 0$  for  $i = 1, \dots, M, j = 1, \dots, N$ .

To take advantages of the finite-difference structure of these operators, we let  $S$  be the iterator of cyclic proximal BCD in Algorithm 3.1. We split  $\{1, 2, \dots, m\}$  into 2 blocks (for case 3) or 4 blocks (for cases 1 and 2), which are inspired by the popular red-black ordering [37] for solving sparse linear system.

According to Theorem 3.5, running finitely many epochs of cyclic proximal BCD gives us a bounded relative error in Def. 3.3. We expect that this solver brings faster overall convergence. Specifically, when  $g^*$  is linear (or equivalently,  $g$  is a  $\delta$  function), the  $z$ -subproblem in PrePDHG reduces to a linear system with a structured sparse matrix  $AA^T$ . Therefore, Gradient Descent amounts to the Richardson method [35, 37], and cyclic proximal BCD is equivalent to the Gauss-Seidel method [17, 37]. The following two claims tell us that  $S$  in Algorithm 3.1 has a closed form, so Algorithm 3.1 is easy to implement. Furthermore, each execution of  $S$  can use parallel computing.

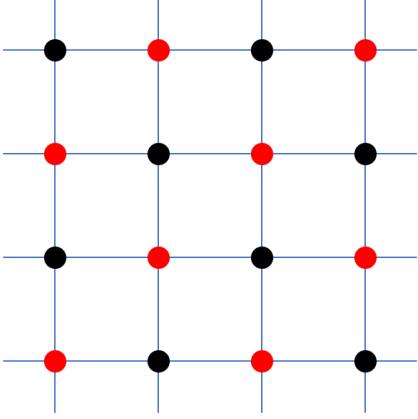


FIG. 1. two-block ordering in Claim 4.1

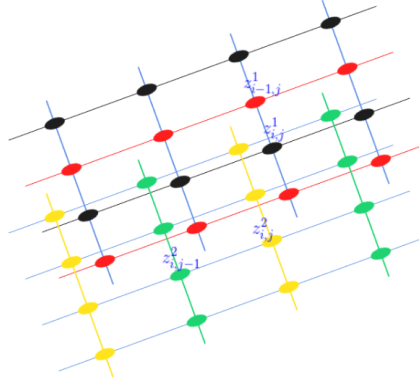


FIG. 2. four-block ordering in Claim 4.2

CLAIM 4.1. When  $A = \text{div}$  (i.e.  $A^T = -D$ ) and  $M_2 = \tau AA^T$ , for  $z \in \mathbb{R}^{M \times N}$ , we separate  $z$  into two block  $z_b, z_r$  where

$$z_b := \{z_{i,j} \mid i + j \text{ is even}\}, \quad z_r := \{z_{i,j} \mid i + j \text{ is odd}\},$$

for  $1 \leq i \leq M, 1 \leq j \leq N$ . If  $g(z) = \sum_{i,j} g_{i,j}(z_{i,j})$  and  $\text{prox}_{\lambda g_{i,j}^*}$  have closed-form solutions for all  $1 \leq i \leq M, 1 \leq j \leq N$  and  $\lambda > 0$ , then  $S$  as the iterator of cyclic proximal BCD in Algorithm 3.1 has a closed form and computing  $S$  is parallelizable.

*Proof.* As illustrated in Fig. 1, every black node is connected to its neighbor red nodes, so we can update all the coordinates corresponding to the black nodes in parallel, while those corresponding to the red nodes are fixed, and vice versa. See Appendix D for a complete explanation.  $\square$

CLAIM 4.2. When  $A = D$  or  $A = D_w$  (i.e.  $A^T = -\text{div}$  or  $A^T = -\text{div diag}(w)$ ) and  $M_2 = \tau AA^T$ , for  $z = (z^1, z^2)^T \in \mathbb{R}^{2M \times N}$ , we separate  $z$  into four blocks  $z_b$ ,  $z_r$ ,  $z_y$  and  $z_g$ , where

$$\begin{aligned} z_b &= \{z_{i,j}^1 \mid i \text{ is odd}\}, & z_r &= \{z_{i,j}^1 \mid i \text{ is even}\}, \\ z_y &= \{z_{i,j}^2 \mid j \text{ is odd}\}, & z_g &= \{z_{i,j}^2 \mid j \text{ is even}\}, \end{aligned}$$

for  $1 \leq i \leq M$ ,  $1 \leq j \leq N$ . If  $g(z) = \sum_{i,j} g_{i,j}(z_{i,j})$  and  $\text{prox}_{\lambda g_{i,j}^*}$  have closed-form solutions for all  $1 \leq i \leq M$ ,  $1 \leq j \leq N$  and  $\lambda > 0$ , then  $S$  as the iterator of cyclic proximal BCD in Algorithm 3.1 has a closed form and computing  $S$  is parallelizable.

*Proof.* In Figure 2, the 4 blocks are in 4 different colors. The coordinates corresponding to nodes of the same color can be updated in parallel, while the rest are fixed. See Appendix D for details.  $\square$

In Table 1, Table 2, Fig. 7, and Table 3, PDHG denotes original PDHG in (3.2) without any preconditioning; DP-PDHG denotes the diagonally-preconditioned PDHG in [32], PrePDHG denotes Preconditioned PDHG in (3.3) where the  $(k+1)$ th  $z$ -subproblem is solved until  $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$  using the TFOCS [4] implementation of FISTA with restart; iPrePDHG (S=BCD) and iPrePDHG (S=FISTA) denote our iPrePDHG in Algorithm 3.1 with the iterator  $S$  being cyclic proximal BCD or FISTA with restart, respectively. All the experiments were performed on MATLAB R2018a on a MacBook Pro with a 2.5 GHz Intel i7 processor and 16GB of 2133MHz LPDDR3 memory.

A comparison between PDHG and DP-PDHG is presented in [32] on TV-L<sup>1</sup> denoising and graph cuts, and in [38] on CT reconstruction. A PDHG algorithm is proposed to estimate earth mover's distance (or optimal transport) in [25]. In order to provide a direct comparison, we use their problem formulations.

**4.1. Total variation based image denoising.** The following problem is known as the (discrete) TV-L<sup>1</sup> model for image denoising:

$$\text{minimize}_u \quad \Phi(u) = \|Du\|_1 + \lambda \|u - b\|_1,$$

where  $D$  is the 2D discrete gradient operator with  $h = 1$ ,  $b \in \mathbb{R}^{M \times N}$  is a noisy input image, and  $\lambda$  is a regularization parameter. In our experiment we input a  $1024 \times 1024$  image with noise level 0.15 and set  $\lambda = 1$ ; see Fig. 3. We run the algorithms until  $\delta^k := \frac{|\Phi^k - \Phi^*|}{|\Phi^*|} < 10^{-6}$ , where  $\Phi^k$  is the objective value at  $k$ th iteration and  $\Phi^*$  is the optimal objective value obtained by calling CVX [11, 20].

Observed performance is summarized in Table 1, where the best results for  $\tau \in \{10, 1, 0.1, 0.01, 0.001\}$  and  $p \in \{1, 2, 3\}$  are presented. Our iPrePDHG (S=BCD) is significantly faster than the other three algorithms.

Remarkably, our algorithm uses fewer outer iterations than PrePDHG under the stopping criterion  $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$ , as this kind of stopping criteria may become looser as  $z^k$  is closer to  $z^*$ . In this example,  $\frac{\|z^k - z^{k+1}\|_2}{\max\{1, \|z^{k+1}\|_2\}} < 10^{-5}$  only requires 1 inner iteration of FISTA when Outer Iter  $\geq 368$ , while as high as 228 inner iterations on average during the first 100 outer iterations. In comparison, our algorithm uses fewer outer iterations while each of them also costs less.

In addition, the diagonal preconditioner given in [32] appears to help very little when  $A = D$ . In fact,  $M_1 = \text{diag}(\sum_i |A_{i,j}|)$  will be  $4I_n$  and  $M_2 = \text{diag}(\sum_j |A_{i,j}|)$  will be

$2I_m$  if we ignore the Neumann boundary condition. Therefore, DP-PDHG performs even worse than PDHG.

Method	Parameters	Outer Iter	Runtime(s)
PDHG	$\tau = 0.01, M_1 = \frac{1}{\tau}I_n, M_2 = \tau\ D\ ^2I_m$	2990	114.2576
DP-PDHG	$M_1 = \text{diag}(\Sigma_i D_{i,j} ), M_2 = \text{diag}(\Sigma_j D_{i,j} )$	8856	329.7890
PrePDHG (ADMM)	$\tau = 0.1, M_1 = \frac{1}{\tau}I_n, M_2 = \tau DD^T$	963	5706.2837
iPrePDHG (S=BCD)	$\tau = 0.01, M_1 = \frac{1}{\tau}I_n, M_2 = \tau DD^T, p = 1$	<b>541</b>	<b>26.2704</b>

TABLE 1

*TV- $L^1$  denoising test. PDHG is original PDHG. DP-PDHG uses diagonal preconditioning. PrePDHG uses non-diagonal preconditioning. iPrePDHG (S=BCD) is our algorithm that uses both non-diagonal preconditioning and an iterator S instead of solving the z-subproblem.*



FIG. 3. Noisy image



FIG. 4. Denoising by iPrePDHG (S=BCD)

**4.2. Graph cuts.** The total-variation-based graph cut model involves minimizing a weighted TV energy:

$$\begin{aligned} & \text{minimize} && \|D_w u\|_1 + \langle u, \omega^u \rangle \\ & \text{subject to} && 0 \leq u \leq 1, \end{aligned}$$

where  $w^u \in \mathbb{R}^{M \times N}$  is a vector of unary weights,  $w^b \in \mathbb{R}^{2MN}$  is a vector of binary weights, and  $D_w = \text{diag}(w^b)D$  for  $D$  being the 2D discrete gradient operator with  $h = 1$ . Specifically, we have  $w_{i,j}^u = \alpha(\|I_{i,j} - \mu_f\|^2 - \|I_{i,j} - \mu_b\|^2)$ ,  $w_{i,j}^{b,1} = \exp(-\beta|I_{i+1,j} - I_{i,j}|)$ , and  $w_{i,j}^{b,2} = \exp(-\beta|I_{i,j+1} - I_{i,j}|)$ . In our experiment, the image has a size  $660 \times 720$ , and we set  $\alpha = 1/2$ ,  $\beta = 10$ ,  $\mu_f = [0; 0; 1]$  (for the blue foreground) and  $\mu_b = [0; 1; 0]$  (for the green background). We run all algorithms until  $\delta^k := \frac{|\Phi^k - \Phi^*|}{|\Phi^*|} < 10^{-8}$ , where  $\Phi^k$  is the objective value at the  $k$ th iteration and  $\Phi^*$  is the optimal objective value obtained by running CVX.

The best results of  $\tau \in \{10, 1, 0.1, 0.01, 0.001\}$  and  $p \in \{1, 2, 3\}$  are summarized in Table 2, where we can see that our iPrePDHG (S=BCD) is the fastest. It is also worth mentioning that its number of outer iterations is close to that of PrePDHG, which solves  $z$ -subproblem much more accurately.



Method	Parameters	Outer Iter	Runtime(s)
PDHG	$\tau = 1, M_1 = \frac{1}{\tau}I_n, M_2 = \tau\ D_w\ ^2I_m$	5529	140.5777
DP-PDHG	$M_1 = \text{diag}(\Sigma_i D_{w_i,j} ),$ $M_2 = \text{diag}(\Sigma_j D_{w_i,j} )$	3571	104.5392
PrePDHG (ADMM)	$\tau = 10, M_1 = \frac{1}{\tau}I_n, M_2 = \tau D_w D_w^T$	282	938.3787
iPrePDHG (S=BCD)	$\tau = 10, M_1 = \frac{1}{\tau}I_n, M_2 = \tau D_w D_w^T, p = 2$	<b>411</b>	<b>14.9663</b>

TABLE 2  
Graph cut test



FIG. 5. Input image



FIG. 6. Graph cut by iPrePDHG (S=BCD)

**4.3. Earth mover's distance.** Earth mover's distance is useful in image processing, computer vision, and statistics [23, 28, 31]. A recent method [25] to compute earth mover's distance is based on

$$(4.2) \quad \begin{aligned} & \text{minimize} && \|m\|_{1,2} \\ & \text{subject to} && \text{div}(m) + \rho^1 - \rho^0 = 0, \end{aligned}$$

where  $m \in \mathbb{R}^{2M \times N}$  is the sought flux vector on the  $M \times N$  grid, and  $\rho^0, \rho^1$  represents two mass distributions on the  $M \times N$  grid. The setting in our experiment here is the same with that in [25], i.e.  $M = N = 256$ ,  $h = \frac{N-1}{4}$ , and for  $\rho^0$  and  $\rho^1$  see Fig. 8.

Since the iterates  $m^k$  may not satisfy the linear constraint, the objective  $\Phi(m) = I_{\{m|\text{div}(m)=\rho^0-\rho^1\}} + \|m\|_{1,2}$  is not comparable. Instead, we compare  $\|m^k\|_{1,2}$  and the constraint violation until  $k = 100000$  outer iterations in Fig. 7, where we set  $\tau = 3 \times 10^{-6}$  as in [25], and  $\sigma = \frac{1}{\tau\|\text{div}\|^2}$ . In Fig. 7, we can see that our algorithm provides much lower constraint violation and much more faithful earth mover's distance  $\|m\|_{1,2}$ . Fig. 8 shows the solution obtained by our iPrePDHG (S=BCD), where  $m$  is the flux that moves the standing cat  $\rho^1$  into the crouching cat  $\rho^0$ . DP-PDHG and PrePDHG are extremely slow in this example. Similar to 4.1, when  $A = \text{div}$ , the diagonal preconditioners proposed in [32] are approximately equivalent to fixed constant parameters  $\tau = \frac{1}{2h}$ ,  $\sigma = \frac{1}{4h}$  and they lead to extremely slow convergence. As for PrePDHG, it suffers from the high cost per outer iteration.

It is worth mentioning that unlike [25], the algorithms in our experiments are not parallelized. On the other hand, in our iPrePDHG (S=BCD), iterator  $S$  can be parallelized (which we did not implement). Therefore, one can expect a further speedup by a parallel implementation.

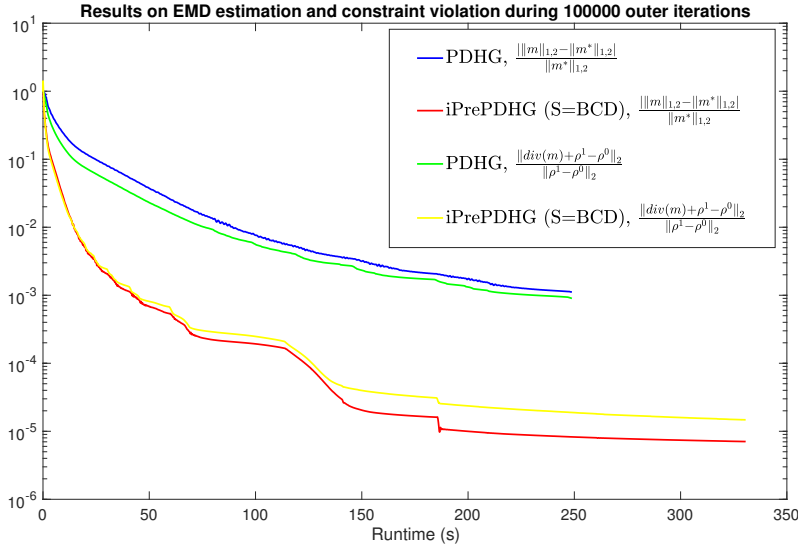


FIG. 7. For PDHG,  $\tau = 3 \times 10^{-6}$ ,  $\sigma = \frac{1}{\tau \|\text{div}\|^2}$ ; For iPrePDHG (S=BCD),  $\tau = 3 \times 10^{-6}$ ,  $M_1 = \tau^{-1} I_n$ ,  $M_2 = \tau \text{divdiv}^T$ ,  $p = 2$ .  $\|m^*\|_{1,2}$  is obtained by calling CVX.



FIG. 8.  $\rho^0$ ,  $\rho^1$  are the white standing cat and the black crouching cat, respectively. Both images are  $256 \times 256$ , and the earth mover's distance between  $\rho^0$  and  $\rho^1$  is 0.6718.

**4.4. CT reconstruction.** We test solving the following optimization problem for CT image reconstruction:

$$(4.3) \quad \text{minimize} \quad \Phi(u) = \frac{1}{2} \|Ru - b\|_2^2 + \lambda \|Du\|_1,$$

where  $R \in \mathbb{R}^{13032 \times 65536}$  is a system matrix for 2D fan-beam CT with a curved detector,  $b = Ru_{\text{true}} \in \mathbb{R}^{13032}$  is a vector of line-integration values, and we want to reconstruct  $u_{\text{true}} \in \mathbb{R}^{MN}$ , where  $M = N = 256$ .  $D$  is the 2D discrete gradient operator with  $h = 1$ , and  $\lambda = 1$  is a regularization parameter. By using the *fancurvedtomo* function from the AIR Tools II [22] package, we generate a test problem where the projection angles are  $0^\circ, 10^\circ, \dots, 350^\circ$ , and for all the other input parameters we use the default values.

Following [38], we formulate the problem (4.3) in the form of (1.1) by taking

$$(4.4) \quad g \begin{pmatrix} p \\ q \end{pmatrix} = \frac{1}{2} \|p - b\|_2^2 + \lambda \|q\|_1, \quad f(u) = 0, \quad A = \begin{pmatrix} R \\ D \end{pmatrix},$$

By using this formulation, we avoid inverting the matrices  $R$  and  $D$ .

Since the block structure of  $AA^T$  is rather complicated, if we naively choose  $M_1 = \frac{1}{\tau} I_n$  and  $M_2 = \tau AA^T$  like in the previous three experiments, it becomes hard to find a fast subproblem solver for the  $z$ -subproblem. In Table 3, we report a TFOCS implementation of FISTA for solving the  $z$ -subproblem and the overall convergence is very slow.

Instead, we propose to choose

$$(4.5) \quad M_1 = \frac{2}{\tau} I_n, \quad M_2 = \begin{pmatrix} \tau \|R\|^2 I_{m-2n} & 0 \\ 0 & \tau DD^T \end{pmatrix}$$

or

$$(4.6) \quad M_1 = \text{diag}(\Sigma_i |R_{i,j}|) + \frac{1}{\tau} I_n, \quad M_2 = \begin{pmatrix} \text{diag}(\Sigma_j |R_{i,j}|) & 0 \\ 0 & \tau DD^T \end{pmatrix}.$$

These choices satisfy (3.5), and have simple block structures, a fixed epoch of  $S$  as cyclic proximal BCD iterations gives fast overall convergence. Note that (4.6) is a little slower but avoids the need of estimating  $\|R\|$ .

We summarize the numerical results in Table 3. All the algorithms are executed until  $\delta^k := \frac{|\Phi^k - \Phi^*|}{|\Phi^*|} < 10^{-4}$ , where  $\Phi^k$  is the objective value at the  $k$ th iteration and  $\Phi^*$  is the optimal objective value obtained by calling CVX. The best results of  $\tau \in \{10, 1, 0.1, 0.01, 0.001\}$  and  $p \in \{1, 2, 3\}$  are summarized in Table 3, for iPrePDHG (S=FISTA) with  $M_2 = \tau AA^T$ , the result for  $p = 100$  is also reported (here we use the TFOCS implementation of FISTA).

**5. Conclusion.** We have developed an approach to accelerate PDHG and ADMM in this paper. Our approach uses effective preconditioners to significantly reduce the number of iterations. In general, most effective preconditioners are non-diagonal and cause very difficult subproblems in PDHG and ADMM, so previous arts are restrictive with less effective diagonal preconditioners. However, we deal with those difficult subproblems by “solving” them highly inexactly, running just very few epochs of proximal BCD iterations. In all of our numerical tests, our algorithm needs relatively few outer iterations (due to effective preconditioners) and has the shortest total running time, achieving 7–95 times speedup over the next best algorithm.

Theoretically, we show a fixed number of inner iterations suffice for global convergence though a new relative error condition. The number depends on various factors but is easy to choose in all of our numerical results.

There are still open questions left for us to address in the future: (a) Depending on problem structures, there are choices of preconditioners that are better than  $M_1 = \frac{1}{\tau} I_n, M_2 = \tau AA^T$  (the ones that lead to ADMM if the subproblems are solved

Method	Parameters	Outer Iter	Runtime(s)
PDHG	$\tau = 0.001, M_1 = \frac{1}{\tau}I_n, M_2 = \tau\ A\ ^2I_m$	364366	3663.0348
DP-PDHG	$M_1 = \text{diag}(\Sigma_i A_{i,j} ),$ $M_2 = \text{diag}(\Sigma_j A_{i,j} )$	70783	713.9865
PrePDHG (ADMM)	$\tau = 0.01, M_1 = \frac{1}{\tau}I_n, M_2 = \tau AA^T$	-	$> 10^4$
iPrePDHG (S=FISTA)	$\tau = 0.001, M_1 = \frac{1}{\tau}I_n,$ $M_2 = \tau AA^T, p = 1, 2, \text{ or } 3$	-	$> 10^4$
iPrePDHG (S=FISTA)	$\tau = 0.01, M_1 = \frac{1}{\tau}I_n,$ $M_2 = \tau AA^T, p = 100$	-	$> 10^4$
iPrePDHG (S=BCD)	$\tau = 0.01, M_1 = \frac{2}{\tau}I_n,$ $M_2 = \begin{pmatrix} \tau\ R\ ^2 & 0 \\ 0 & \tau DD^T \end{pmatrix}, p = 2$	<b>587</b>	<b>7.5365</b>
iPrePDHG (S=BCD)	$\tau = 0.01, M_1 = \text{diag}(\Sigma_i R_{i,j} ) + \frac{1}{\tau}I_n,$ $M_2 = \begin{pmatrix} \text{diag}(\Sigma_j R_{i,j} ) & 0 \\ 0 & \tau DD^T \end{pmatrix}, p = 2$	<b>858</b>	<b>10.3517</b>

TABLE 3  
CT reconstruction

exactly). For example, in CT reconstruction, our choices of  $M_1$  and  $M_2$  have much faster overall convergence. (b) Is it possible to show Algorithm 3.1 converges even with  $S$  chosen as the iterator of faster accelerated solvers like APCG [26], NU\_ACDM [1], and A2BCD [21]? (c) In general, how to accelerate a broader class of algorithms by integrating effective preconditioning and cheap inner loops while still ensuring global convergence?

### Appendix A. ADMM as a special case of PrePDHG.

In this section we show that if we choose  $M_1 = \frac{1}{\tau}$  and  $M_2 = \tau AA^T$  in PrePDHG (3.3), then it is equivalent to ADMM on the primal problem (1.1).

By Theorem 1 of [41], we know that ADMM is primal-dual equivalent, in the sense that one can recover primal iterates from dual iterates and vice versa. Therefore, it suffices to show that  $M_1 = \frac{1}{\tau}$  and  $M_2 = \tau AA^T$  in PrePDHG (3.3) on the primal problem is equivalent to ADMM on the dual problem (1.2).

In Theorem 3.6 we have shown that, under an appropriate change of variables, PrePDHG on the primal is equivalent to applying (3.15) to the dual. As a result, we just need to demonstrate that the latter is exactly ADMM on the dual when  $M_1 = \frac{1}{\tau}I_{n \times n}$  and  $M_2 = \tau AA^T$ .

For the  $z$ -update in (3.15), we have

$$\begin{aligned}
z^{k+1} &= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) - \tau \langle z - z^k, A(-A^T z^k - y^k + u^k) \rangle + \frac{\tau}{2} \|z - z^k\|_{AA^T}^2\} \\
&= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) - \tau \langle z - z^k, A(-y^k + u^k) \rangle + \frac{\tau}{2} \|z\|_{AA^T}^2\} \\
&= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) + \tau \langle z, A(y^k - u^k) \rangle + \frac{\tau}{2} \|A^T z\|^2\} \\
&= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) + \tau \langle A^T z, -u^k \rangle + \frac{\tau}{2} \|A^T z + y^k\|^2\} \\
\text{(A.1)} \quad &= \arg \min_{z \in \mathbb{R}^m} \{g^*(z) + \tau \langle -A^T z - y^k, u^k \rangle + \frac{\tau}{2} \|A^T z + y^k\|^2\}.
\end{aligned}$$

and for the  $y$ -update we have

$$\begin{aligned}
y^{k+1} &= \text{Prox}_{f^*}^{M^{-1}}(u^k - A^T z^{k+1}) \\
&= \arg \min_{y \in \mathbb{R}^n} \left\{ f^*(y) + \frac{\tau}{2} \|y - u^k + A^T z^{k+1}\|^2 \right\} \\
\text{(A.2)} \quad &= \arg \min_{y \in \mathbb{R}^n} \left\{ f^*(y) + \tau \langle -A^T z^{k+1} - y, u^k \rangle + \frac{\tau}{2} \|A^T z^{k+1} + y\|^2 \right\}.
\end{aligned}$$

Define  $v^k = \tau u^k$ , (A.1), (A.2), and the  $u$ -update in (3.15) become

$$\begin{aligned}
z^{k+1} &= \arg \min_{z \in \mathbb{R}^m} \left\{ g^*(z) + \langle -A^T z - y^k, v^k \rangle + \frac{\tau}{2} \|A^T z + y^k\|^2 \right\}, \\
y^{k+1} &= \arg \min_{y \in \mathbb{R}^n} \left\{ f^*(y) + \langle -A^T z^{k+1} - y, v^k \rangle + \frac{\tau}{2} \|A^T z^{k+1} + y\|^2 \right\}, \\
v^{k+1} &= v^k - \tau(A^T z^{k+1} + y^{k+1}),
\end{aligned}$$

which are ADMM iterations on the dual problem (1.2).

**Appendix B. Proof of Theorem 3.5: bounded relative error when  $S$  is the iterator of cyclic proximal BCD.**

The  $z$ -subproblem in (3.3) has the form

$$\min_{z \in \mathbb{R}^m} h_1(z) + h_2(z),$$

where  $h_1(z) = g^*(z) = \sum_{i=1}^l g_i^*(z_i)$ , and  $h_2(z) = \frac{1}{2} \|z - z^k - M_2^{-1} A(2x^{k+1} - x^k)\|_{M_2}^2$ . And  $z^{k+1} = z_p^{k+1}$  is given by

$$\begin{aligned}
z_0^{k+1} &= z^k, \\
z_{i+1}^{k+1} &= S(z_i^{k+1}, x^{k+1}, x^k), \quad i = 0, 1, \dots, p-1,
\end{aligned}$$

Here,  $S$  is the iterator of cyclic proximal BCD. Define

$$\begin{aligned}
T(z) &= \text{Prox}_{\gamma g^*(z)}(z - \gamma \nabla h_2(z)), \\
B(z) &= \frac{1}{\gamma}(z - T(z)),
\end{aligned}$$

and the  $i$ th coordinate operator of  $B$ :

$$B_i(z) = (0, \dots, (B(z))_i, \dots, 0).$$

Then, we have

$$z_{i+1}^{k+1} = S(z_i^{k+1}, x^{k+1}, x^k) = (I - \gamma B_l)(I - \gamma B_2) \dots (I - \gamma B_1) z_i^{k+1}.$$

By [3, Prop. 26.16(ii)], we know that  $T(z)$  is a contraction with coefficient  $\theta = \sqrt{1 - \gamma(2\lambda_{\min}(M_2) - \gamma\lambda_{\max}^2(M_2))}$ . We know that for  $\forall z_1, z_2 \in \mathbb{R}^m$  and  $\mu = \frac{1-\theta}{\gamma}$ ,

$$\begin{aligned}
\langle B(z_1) - B(z_2), z_1 - z_2 \rangle &= \frac{1}{\gamma} \|z_1 - z_2\|^2 - \frac{1}{\gamma} \langle T(z_1) - T(z_2), z_1 - z_2 \rangle \\
&\geq \mu \|z_1 - z_2\|^2,
\end{aligned}$$

Let  $z_\star^{k+1} = \arg \min_{z \in \mathbb{R}^m} \{h_1(z) + h_2(z)\}$ . For [9, Thm 3.5], we have

$$(B.1) \quad \|z_i^{k+1} - z_\star^{k+1}\| \leq \rho^i \|z_0^{k+1} - z_\star^{k+1}\|, \quad \forall i = 1, 2, \dots, p.$$

where  $\rho = 1 - \frac{\gamma\mu^2}{2}$ .

Let  $y_j = (I - \gamma B_j) \dots (I - \gamma B_1) z_{p-1}^{k+1}$  for  $j = 1, \dots, l$  and  $y_0 = z_{p-1}^{k+1}$ . Note that  $(z_p^{k+1})_j = (y_j)_j$  for  $j = 1, 2, \dots, l$ , and the blocks of  $y_j$  satisfies

$$(y_j)_t = \begin{cases} \left( \text{Prox}_{\gamma g^*} (y_{j-1} - \gamma \nabla h_2(y_{j-1})) \right)_t, & \text{if } t = j \\ (y_{j-1})_t, & \text{otherwise.} \end{cases}$$

On the other hand, we have

$$\text{Prox}_{\gamma g^*} (y_{j-1} - \gamma \nabla h_2(y_{j-1})) = \arg \min_{y \in \mathbb{R}^m} \left\{ g^*(y) + \frac{1}{2\gamma} \|y - y_{j-1} + \gamma \nabla h_2(y_{j-1})\|^2 \right\}.$$

Since  $g^*$  and  $\|\cdot\|^2$  are separable, we obtain

$$\mathbf{0} \in \partial g_j^*((y_j)_j) + \frac{1}{\gamma} \left( (y_j)_j - (y_{j-1})_j + \gamma (\nabla h_2(y_{j-1}))_j \right), \quad \forall j = 1, 2, \dots, l,$$

or equivalently,

$$\mathbf{0} \in \partial g_j^*((z_p^{k+1})_j) + \frac{1}{\gamma} \left( (z_p^{k+1})_j - (z_{p-1}^{k+1})_j + \gamma (\nabla h_2(y_{j-1}))_j \right), \quad \forall j = 1, 2, \dots, l.$$

Therefore,

$$\mathbf{0} \in \partial g^*(z_p^{k+1}) + \frac{1}{\gamma} \left( z_p^{k+1} - z_{p-1}^{k+1} + \gamma \xi_p \right), \quad \forall j = 1, 2, \dots, l,$$

where  $(\xi_p)_j = (\nabla h_2(y_{j-1}))_j$  for  $j = 1, 2, \dots, l$ . Comparing this with (3.7), we obtain

$$\varepsilon^{k+1} = \xi_p - \nabla h_2(z_p^{k+1}) + \frac{1}{\gamma} (z_p^{k+1} - z_{p-1}^{k+1}).$$

Notice that the first  $j-1$  blocks of  $y_{j-1}$  are the same with those of  $y_l = z_p^{k+1}$ , and the rest of the blocks are the same with those of  $y_0 = z_{p-1}^{k+1}$ , so we have

$$\begin{aligned} \|\varepsilon^{k+1}\| &\leq \sum_{j=1}^l \lambda_{\max}(M_2) \|y_{j-1} - z_p^{k+1}\| + \frac{1}{\gamma} \|z_p^{k+1} - z_{p-1}^{k+1}\| \\ &\leq l \lambda_{\max}(M_2) \|z_{p-1}^{k+1} - z_p^{k+1}\| + \frac{1}{\gamma} \|z_p^{k+1} - z_{p-1}^{k+1}\| \\ &\leq (l \lambda_{\max}(M_2) + \frac{1}{\gamma}) (\|z_p^{k+1} - z_\star^{k+1}\| + \|z_{p-1}^{k+1} - z_\star^{k+1}\|) \end{aligned}$$

Combine this with (B.1)

$$(B.2) \quad \|\varepsilon^{k+1}\| \leq (l \lambda_{\max}(M_2) + \frac{1}{\gamma}) (\rho^p + \rho^{p-1}) \|z_0^{k+1} - z_\star^{k+1}\|.$$

Combining

$$\begin{aligned}\|z^{k+1} - z^k\| &= \|z_p^{k+1} - z_0^{k+1}\| \\ &\geq \|z_0^{k+1} - z_\star^{k+1}\| - \|z_p^{k+1} - z_\star^{k+1}\| \\ &\geq (1 - \rho^p) \|z_0^{k+1} - z_\star^{k+1}\|\end{aligned}$$

with (B.2), we obtain

$$\|\varepsilon^{k+1}\| \leq \frac{(l\lambda_{\max}(M_2) + \frac{1}{\gamma})(\rho^p + \rho^{p-1})}{1 - \rho^p} \|z^{k+1} - z^k\|.$$

### Appendix C. Proof of Theorem 3.10: KŁ property gives global convergence.

According to Theorem 3.6, we just need to show that  $\{M_1^{-1}u^k, z^k\}$  converges to a primal-dual solution pair of (1.1).

By Theorem 3.9, we can take  $\{z^{k_s}, y^{k_s}, u^{k_s}\} \rightarrow (z^c, y^c, u^c)$  as  $s \rightarrow \infty$ . Note that  $L(z^{k_s}, y^{k_s}, u^{k_s})$  is monotonic nonincreasing and lower bounded due to Theorem 3.7, which implies the convergence of  $L(z^{k_s}, y^{k_s}, u^{k_s})$ . Since  $L$  is lower semicontinuous, we have

$$(C.1) \quad L(z^c, y^c, u^c) \leq \lim_{s \rightarrow \infty} L(z^{k_s}, y^{k_s}, u^{k_s}).$$

Since the only potentially discontinuous terms in  $L$  is  $g^*$ , we have

$$(C.2) \quad \lim_{s \rightarrow \infty} L(z^{k_s}, y^{k_s}, u^{k_s}) - L(z^c, y^c, u^c) \leq \limsup_{s \rightarrow \infty} g^*(z^{k_s}) - g^*(z^c).$$

By (3.21), we know that

$$\begin{aligned}g^*(z^c) &\geq g^*(z^{k_s}) \\ &\quad + \langle M_2(z^{k_s-1} - z^{k_s}) + AM_1^{-1}(-A^T z^{k_s-1} - y^{k_s-1} + u^{k_s-1}) - \varepsilon^{k_s}, z^c - z^{k_s} \rangle,\end{aligned}$$

Then, by Theorem 3.7, we further get  $z^{k_s-1} - z^{k_s} \rightarrow \mathbf{0}$ . Since  $z^{k_s} \rightarrow z^c$  and  $\{z^k, y^k, u^k\}$  is bounded, we obtain

$$\limsup_{s \rightarrow \infty} g^*(z^{k_s}) - g^*(z^c) \leq 0.$$

Combining this with (C.1) and (C.2), we conclude that  $\lim_{s \rightarrow \infty} L(z^{k_s}, y^{k_s}, u^{k_s}) = L(z^c, y^c, u^c)$ .

Since  $g^*$  is a KŁ function,  $L$  is also KŁ. Consequently, similar to Theorem 2.9 of [2], we can claim the convergence of  $\{z^k, y^k, u^k\}$  to  $\{z^c, y^c, u^c\}$ .

### Appendix D. Two-block ordering in Claim 4.1 and four-block ordering in Claim 4.2.

According to (3.4), when  $M_2 = \tau AA^T$ , the  $z$ -subproblem of Algorithm 3.1 is

$$(D.1) \quad z^{k+1} = \arg \min_{z \in \mathbb{R}^m} \{g^*(z) - \langle z - z^k, A(2x^{k+1} - x^k) \rangle + \frac{\tau}{2} \|A^T(z - z^k)\|_2^2\}.$$

Let us prove Claim 4.1 first. In that claim,  $A = \text{div} \in \mathbb{R}^{MN \times 2MN}$  and  $z \in \mathbb{R}^{MN}$ . Following the definition of the sets  $z_b$  and  $z_r$ , we separate the  $MN$  columns of  $A^T = -D$  into two blocks  $L_b, L_r$  by associating them with  $z_b$  and  $z_r$ , respectively. Therefore, we have  $A^T z = L_b z_b + L_r z_r$  for any  $z \in \mathbb{R}^{MN}$ .

By the red-black ordering in Fig. 1, different columns of  $L_b$  are orthogonal one another, so  $L_b^T L_b$  is diagonal. Similarly,  $L_r^T L_r$  is also diagonal.

Let  $b$  be the set of black nodes and  $r$  the set of red nodes. We can rewrite (D.1) as

$$(D.2) \quad z^{k+1} = \arg \min_{z_b, z_r \in \mathbb{R}^{MN/2}} \{g_b^*(z_b) + g_r^*(z_r) + \langle z_b + z_r, c^k \rangle + \frac{\tau}{2} \|L_b(z_b - z_b^k) + L_r(z_r - z_r^k)\|_2^2\},$$

where  $g_b^*(z_b) = \sum_{(i,j) \in b} g_{i,j}^*(z_{i,j})$ ,  $g_r^*(z_r) = \sum_{(i,j) \in r} g_{i,j}^*(z_{i,j})$ , and  $c^k = -A(2x^{k+1} - x^k)$ .

Applying cyclic proximal BCD to black and red blocks alternatively yields

$$(D.3) \quad z_b^{k+\frac{t+1}{p}} = \text{prox}^{\tau L_b^T L_b} \left( g_b^*(\cdot) + \langle \cdot, \tau L_b^T L_r(z_r^{k+\frac{t}{p}} - z_r^k) + c_b^k \rangle \right) \left( z_b^{k+\frac{t}{p}} \right),$$

$$(D.4) \quad z_r^{k+\frac{t+1}{p}} = \text{prox}^{\tau L_r^T L_r} \left( g_r^*(\cdot) + \langle \cdot, \tau L_r^T L_b(z_b^{k+\frac{t+1}{p}} - z_b^k) + c_r^k \rangle \right) \left( z_r^{k+\frac{t}{p}} \right).$$

for  $t = 0, 1, \dots, p-1$ , where  $p$  is the number of inner iterations in Algorithm 3.1.

These updates have closed-form solutions since  $L_b^T L_b$  and  $L_r^T L_r$  are diagonal, and all  $\text{prox}_{\lambda g_{i,j}^*}$  are closed-form. Furthermore, the updates within each block can be done in parallel.

The proof of Claim 4.2 is similar. When  $A = D$  or  $A = D_w$ , we separate the columns of  $A^T$  into four blocks  $L_b, L_r, L_y, L_g$  by associating them with  $z_b, z_r, z_y, z_g$ , respectively. Therefore, we have  $A^T z = L_b z_b + L_r z_r + L_y z_y + L_g z_g$  for all  $z \in \mathbb{R}^{2MN}$ . Similarly, by the block design in Fig. 2, cyclic proximal BCD iterations have closed-form solutions, and updates within each block can be executed in parallel.

#### REFERENCES

- [1] Z. ALLEN-ZHU, Z. QU, P. RICHTÁRIK, AND Y. YUAN, *Even faster accelerated coordinate descent using non-uniform sampling*, in International Conference on Machine Learning, 2016, pp. 1110–1119.
- [2] H. ATTOUCH, J. BOLTE, AND B. F. SVAITER, *Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Mathematical Programming, 137 (2013), pp. 91–129.
- [3] H. H. BAUSCHKE, P. L. COMBETTES, ET AL., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, vol. 2011, Springer, 2017.
- [4] S. R. BECKER, E. J. CANDÈS, AND M. C. GRANT, *Templates for convex cone problems with applications to sparse signal recovery*, Mathematical programming computation, 3 (2011), p. 165.
- [5] K. BREDIES AND H. SUN, *Preconditioned Douglas-Rachford splitting methods for convex-concave saddle-point problems*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 421–444.
- [6] K. BREDIES AND H. SUN, *A proximal point analysis of the preconditioned alternating direction method of multipliers*, Journal of Optimization Theory and Applications, 173 (2017), pp. 878–907.
- [7] A. CHAMBOLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of mathematical imaging and vision, 40 (2011), pp. 120–145.
- [8] A. CHAMBOLE AND T. POCK, *On the ergodic convergence rates of a first-order primal-dual algorithm*, Mathematical Programming, 159 (2016), pp. 253–287.
- [9] Y. T. CHOW, T. WU, AND W. YIN, *Cyclic coordinate-update algorithms for fixed-point problems: Analysis and applications*, SIAM Journal on Scientific Computing, 39 (2017), pp. A1280–A1300.
- [10] P. L. COMBETTES AND N. N. REYES, *Moreau's decomposition in banach spaces*, Mathematical Programming, 139 (2013), pp. 103–114.



- [11] I. CVX RESEARCH, *CVX: Matlab software for disciplined convex programming, version 2.0*. <http://cvxr.com/cvx>, Aug. 2012.
- [12] J. ECKSTEIN AND D. P. BERTSEKAS, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, *Mathematical Programming*, 55 (1992), pp. 293–318.
- [13] J. ECKSTEIN AND W. YAO, *Approximate ADMM algorithms derived from lagrangian splitting*, *Computational Optimization and Applications*, 68 (2017), pp. 363–405.
- [14] J. ECKSTEIN AND W. YAO, *Relative-error approximate versions of Douglas-Rachford splitting and special cases of the ADMM*, *Mathematical Programming*, (2017), pp. 1–28.
- [15] E. ESSER, X. ZHANG, AND T. F. CHAN, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, *SIAM Journal on Imaging Sciences*, 3 (2010), pp. 1015–1046.
- [16] D. FELJER AND F. PAGANINI, *Stability of primal–dual gradient dynamics and applications to network optimization*, *Automatica*, 46 (2010), pp. 1974–1981.
- [17] C. F. GAUSS, *Werke (in German)*, 9, Göttingen: Königlichen Gesellschaft der Wissenschaften, 1903.
- [18] P. GISELSSON AND S. BOYD, *Diagonal scaling in Douglas-Rachford splitting and ADMM*, in *Decision and Control (CDC)*, 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 5033–5039.
- [19] P. GISELSSON AND S. BOYD, *Linear convergence and metric selection for Douglas-Rachford splitting and ADMM*, *IEEE Transactions on Automatic Control*, 62 (2017), pp. 532–544.
- [20] M. GRANT AND S. BOYD, *Graph implementations for nonsmooth convex programs*, in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, eds., *Lecture Notes in Control and Information Sciences*, Springer-Verlag Limited, 2008, pp. 95–110. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [21] R. HANNAH, F. FENG, AND W. YIN, *A2BCD: An asynchronous accelerated block coordinate descent algorithm with optimal complexity*, arXiv preprint arXiv:1803.05578, (2018).
- [22] P. C. HANSEN AND J. S. JØRGENSEN, *Air tools ii: algebraic iterative reconstruction methods, improved implementation*, *Numerical Algorithms*, 79 (2018), pp. 107–137.
- [23] E. LEVINA AND P. BICKEL, *The earth mover’s distance is the mallows distance: Some insights from statistics*, in null, *IEEE*, 2001, p. 251.
- [24] M. LI, L.-Z. LIAO, AND X. YUAN, *Inexact alternating direction methods of multipliers with logarithmic-quadratic proximal regularization*, *Journal of Optimization Theory and Applications*, 159 (2013), pp. 412–436.
- [25] W. LI, E. K. RYU, S. OSHER, W. YIN, AND W. GANGBO, *A parallel method for earth mover’s distance*, *UCLA Comput. Appl. Math. Pub.(CAM) Rep*, (2017), pp. 17–12.
- [26] Q. LIN, Z. LU, AND L. XIAO, *An accelerated proximal coordinate gradient method*, in *Advances in Neural Information Processing Systems*, 2014, pp. 3059–3067.
- [27] Q. LIN, Z. LU, AND L. XIAO, *An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization*, *SIAM Journal on Optimization*, 25 (2015), pp. 2244–2273.
- [28] L. MÉTIVIER, R. BROSSIER, Q. MÉRIGOT, E. OUDET, AND J. VIRIEUX, *Measuring the misfit between seismograms using an optimal transport distance: application to full waveform inversion*, *Geophysical Supplements to the Monthly Notices of the Royal Astronomical Society*, 205 (2016), pp. 345–377.
- [29] M. K. NG, F. WANG, AND X. YUAN, *Inexact alternating direction methods for image recovery*, *SIAM Journal on Scientific Computing*, 33 (2011), pp. 1643–1668.
- [30] B. O’DONOGHUE AND E. CANDÈS, *Adaptive restart for accelerated gradient schemes*, *Foundations of computational mathematics*, 15 (2015), pp. 715–732.
- [31] O. PELE AND M. WERMAN, *Fast and robust earth mover’s distances.*, in *ICCV*, vol. 9, 2009, pp. 460–467.
- [32] T. POCK AND A. CHAMBOLLE, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in *Computer Vision (ICCV)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 1762–1769.
- [33] T. POCK, D. CREMERS, H. BISCHOF, AND A. CHAMBOLLE, *An algorithm for minimizing the Mumford-Shah functional*, in *Computer Vision*, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1133–1140.
- [34] J. RASCH AND A. CHAMBOLLE, *Inexact first-order primal-dual algorithms*, arXiv preprint arXiv:1803.10576, (2018).
- [35] L. F. RICHARDSON, *ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, *Phil. Trans. R. Soc. Lond. A*, 210 (1911), pp. 307–357.

- [36] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Variational Analysis*, vol. 317, Springer Science & Business Media, 2009.
- [37] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, vol. 82, SIAM, 2003.
- [38] E. Y. SIDKY, J. H. JØRGENSEN, AND X. PAN, *Convex optimization problem prototyping for image reconstruction in computed tomography with the chambolle-pock algorithm*, *Physics in Medicine & Biology*, 57 (2012), p. 3065.
- [39] T. VALKONEN, *A primal-dual hybrid gradient method for nonlinear operators with applications to mri*, *Inverse Problems*, 30 (2014), p. 055012.
- [40] Y. WANG, W. YIN, AND J. ZENG, *Global convergence of ADMM in nonconvex nonsmooth optimization*, arXiv preprint arXiv:1511.06324, (2015).
- [41] M. YAN AND W. YIN, *Self equivalence of the alternating direction method of multipliers*, in *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, 2016, pp. 165–194.
- [42] X. ZHANG, M. BURGER, AND S. OSHER, *A unified primal-dual algorithm framework based on bregman iteration*, *Journal of Scientific Computing*, 46 (2011), pp. 20–46.
- [43] M. ZHU AND T. CHAN, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA CAM Report, 34 (2008).